Using theory and experiments of spheres moving near boundaries to optimize the method of images for regularized Stokeslets

Hoa Nguyen,[1,*] Amelia Gibbs,[1,†] Frank Healy,[2,‡] Orrin Shindell,[3,§] Ricardo Cortez,[4,‖] Kathleen M. Brown,[5,¶] Jonathan McCoy,[5,#] and Bruce Rodenborn[5,**]

[1]*Department of Mathematics, Trinity University, San Antonio, Texas 78212, USA*
[2]*Department of Biology, Trinity University, San Antonio, Texas 78212, USA*
[3]*Department of Physics and Astronomy, Trinity University, San Antonio, Texas 78212, USA*
[4]*Department of Mathematics, Tulane University, New Orleans, Louisiana 70118, USA*
[5]*Physics Program, Centre College of Kentucky, Danville, Kentucky 40422, USA*

(Received 10 July 2024; accepted 19 February 2025; published 12 March 2025)

The general system of images for regularized Stokeslets (GSIRS) developed by Cortez and Varela [J. Comput. Phys. **285**, 41 (2015)] is used extensively to model Stokes flow phenomena such as microorganisms swimming near a boundary. Our collaborative team uses dynamically similar scaled macroscopic experiments to test theories for forces and torques on spheres moving near a boundary and uses these data and the method of regularized Stokeslets (MRS) created by Cortez *et al.* [Phys. Fluids **17**, 031504 (2005)] to calibrate the GSIRS. We find excellent agreement between theory and experiments, which provides experimental validation of exact series solutions for spheres moving near an infinite plane boundary. We test two surface discretization methods commonly used in the literature: the 6-patch method and the spherical centroidal Voronoi tessellation (SCVT) method. Our data show that a discretization method, such as SCVT, that uniformly distributes points provides the most accurate results when the motional symmetry is broken by the presence of a boundary. We use theory and the MRS to find optimal values for the regularization parameter in free space for a given surface discretization and show that the optimal regularization parameter values can be fit with simple formulas when using the SCVT method. We also present a regularization function with higher-order accuracy when compared with the regularization function previously introduced by Cortez *et al.* [Phys. Fluids **17**, 031504 (2005)]. The simulated force and torque values compare very well with experiments and theory for a wide range of boundary distances. However, we find that for a fixed discretization of the sphere, the simulations lose accuracy when the gap between the edge of the sphere and the wall is smaller than the average distance between discretization points in the SCVT method. We also show an alternative method to calibrate

*Contact author: hnguyen5@trinity.edu
†Contact author: tgibbs@trinity.edu
‡Contact author: fhealy@trinity.edu
§Contact author: oshindel@trinity.edu
‖Contact author: rcortez@tulane.edu
¶Contact author: kate.brown@centre.edu
#Contact author: jonathan.mccoy@centre.edu
**Contact author: bruce.rodenborn@centre.edu

the GSIRS to simulate sphere motion arbitrarily close to the boundary. Our computational parameters and methods along with our matlab and python implementations of the series solution of Lee and Leal [J. Fluid Mech. **98**, 193 (1980)], MRS, and GSIRS provide researchers with important resources to optimize the GSIRS and other numerical methods, so that they can efficiently and accurately simulate spheres moving near a boundary.

DOI: 10.1103/PhysRevFluids.10.033101

BACKGROUND

The method of regularized Stokeslets (MRS) [1] and the general system of images for regularized Stokeslets (GSIRS) [2] have been used extensively to model swimming microorganisms. The work of Shindell *et al.* [3] showed how dynamically similar macroscopic experiments and theory provide a principled method to calibrate the MRS and GSIRS. They considered a cylinder and helix moving near a boundary and showed that the optimal regularization parameter depended on both the surface discretization and geometry. Accurate forces and torques on a model bacterium could then be calculated and swimming performance measures such as the Purcell efficiency [4], energy per distance [5,6], and metabolic energy cost (energy per distance per body mass) [3] could be computed. Thus, accurately calibrated models allow determination of important structure-function relationships when microorganisms are moving in their environment.

We use similar techniques to calibrate the GSIRS for spherical geometries, which can be used to model *Rhodobacter sphaeroides* and *Enterococcus saccharolyticus* and other microorganisms with round bodies [7–9]. The theoretical analysis of forces and torques on spheres moving near a boundary in Stokes flow has a long history beginning with the work of Jeffrey [10] who first found a series solution to the Stokes equations to calculate the torque on a sphere rotating with its axis perpendicular to an infinite plane. Subsequent work found the drag on a sphere moving parallel to and perpendicular to a boundary [11,12], and the torque on a sphere with its rotation axis oriented parallel to the boundary [13]. The linearity of the Stokes solutions allows any arbitrary motion of a sphere relative to a boundary to be decomposed into a sum of these four fundamental motions [14].

Work by Lee and Leal [14] unified these various series solutions into one linear system of equations that calculates the force and torque for motion near surfaces with varying properties. However, there has been no comprehensive comparison of experiments to theory to validate the drag and torque for the four fundamental motions [14]. Previous experimental work that has been compared to theory has generally focused on sedimentation experiments, such as those by Malysa and van de Ven [15]. They showed good agreement between theory and experiment for spheres moving parallel to a plane wall, but tested no other motions, instead focusing on the small rotation predicted by the theory for this motion. Dufresne *et al.* [16] used microscopic measurements of pairs of colloidal particles moving near a boundary, and their work was also different because the particles were affected by thermal motion. Pitois *et al.* [17] used soap films to create quasi-free-slip surfaces in channel flow experiments and compared their results to Lee and Leal's theory for this boundary type but were focused on the difference between free-slip and no-slip boundaries in channel flow. Wang *et al.* used optical tweezers and thermal noise to measure the drag increase for motion parallel to free-slip boundaries, but did not have the precision that we achieve in our experiments [18]. Likewise, Wu *et al.* used the theory of Brenner [12] in their analysis but were focused on finding attractors in a plane Couette flow setup rather than testing the theory directly [19]. There have been fewer experimental tests of torque on a sphere, such as those by Maxworthy [20] and Kunesh *et al.* [21], but they studied a sphere in a rotating cylindrical tank rather than a rotating sphere in a stationary tank, so the comparison was with a similar but different theory. Thus, there have been no direct tests of the increased torque on a sphere by a plane wall as predicted by Lee and Leal's theory.

Our dynamically similar experiments, therefore, provide a complete test of all four fundamental sphere motions near a boundary. We test the theory over a much larger range of boundary distances,

with higher resolution in boundary distances, and with much greater precision than has been previously reported. Our experiments also use an approach to eliminate finite size effects by using Lee and Leal's theory (user-friendly matlab and python GUI codes are provided in [41]) to calculate the effect of all six boundaries in our tank for a given motion relative to a boundary: five solid boundaries and one free surface. We assume that the effects of the six boundaries add linearly even though the theory is for infinite plane walls and subtract the predicted effects of five of the boundaries from our data. We can thereby compare our experimental measurements to the predicted effect of a single boundary, which would otherwise be significantly affected by other boundaries in some cases (see Sec. I below for more details).

We use the theory and experiments to precisely calibrate our GSIRS models, which include two regularization functions: the commonly used ϕ_ϵ developed by Cortez *et al.* [2] and a new regularization function, ψ_ϵ. We also use two methods of surface discretization: spherical centroidal Voronoi tessellation (SCVT), as developed by Du *et al.* [22], and the 6-patch method [23]. We find optimal regularization parameters by minimizing the percent error between simulations and theory in free space and show that ψ_ϵ improves the order of accuracy in the GSIRS. We also show that the SCVT outperforms the 6-patch discretization scheme because it is less sensitive to the orientation of the discretization to the boundary. However, we find that even the optimized GSIRS loses accuracy when the gap between the sphere and the boundary is smaller than the average distance between Stokeslets on the sphere. This result provides researchers with a new general rule of thumb: the GSIRS may not be accurate within this distance and higher resolution is needed. Our library of experimental measurements and numerical implementations of theory are also reference values to aid researchers in precisely calibrating other numerical methods.

This article is organized as follows: Section I describes our dynamically similar experimental techniques to obtain the forces and torques for the four fundamental motions including removing finite size effects using theory. Section II describes our discretization and regularization schemes including the regularization function ψ_ϵ along with how we optimize the model using theory. Section III compares the experiments and optimized simulations to theory and discuss how to increase accuracy very near the boundary. Section IV summarizes the performance of our calibrated model and our new empirical rule for establishing the minimum discretization needed for accuracy near a boundary. Appendixes A and B provide additional details about the simulations and our implementation of the theory.

I. EXPERIMENTAL METHODS

We used dynamically similar macroscopic tank experiments to test the four fundamental motions of a sphere near an infinite plane: perpendicular translation, parallel translation, perpendicular axis rotation, and parallel axis rotation [14]. Experiments were performed in a 225-liter tank (0.61 m×0.61 m×0.61 m) filled with approximately 180 liters of silicone oil [24]. The fluid has a density of 970 kg/m^3 and dynamic viscosity of $\mu = 51.78 \pm 0.32$ Pa s at 25 °C. The viscosity was measured using a rheometer [25]. The fluid temperature was measured each day of experimental trials and the manufacturer's temperature coefficient for the fluid (1.00×10^{-6} kg/(m s)/°C) was used to adjust the viscosity value. The high-viscosity fluid ($>10^4$ times the viscosity of water) ensures that the Reynolds number was much less than unity in all experiments (Re $\approx 10^{-3}$) so that the incompressible Stokes equations (1) were valid for the typical length and speed scales in the experiments.

We used Lee and Leal's theory to remove finite size effects in the experiments, which was important because the scaled equivalent size of our tank would be about 10 μm and using 0.02-μm spheres if the working fluid was water. To remove unwanted effects, we calculated the predicted contribution of all six boundaries in the tank (five solid and one free surface) for a given motion and then isolated the effects of one boundary by subtracting off the predicted effects of the other five boundaries from our data. For example, spheres translating parallel to the vertical wall in Fig. 1 were receding from the free surface above and approaching the solid surface at the bottom

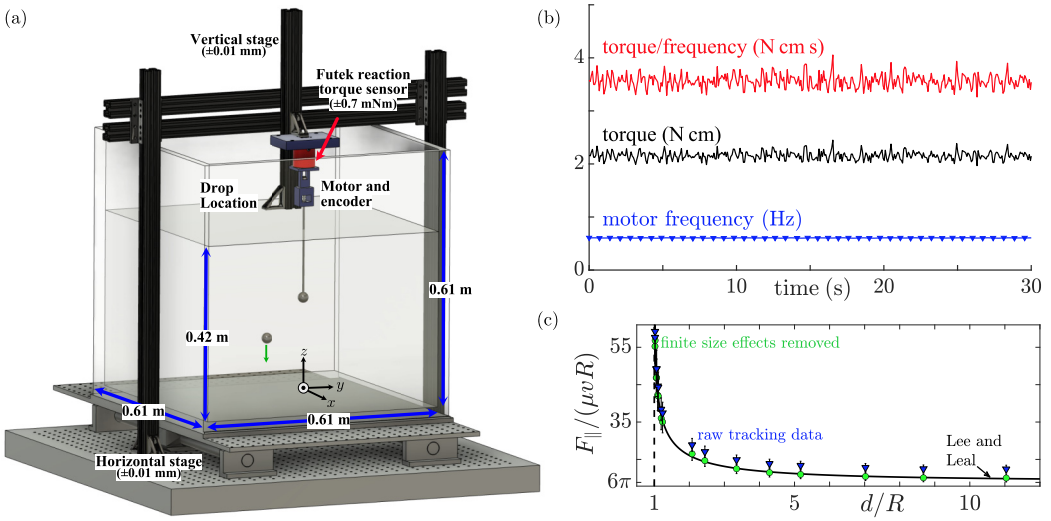

FIG. 1. Experimental setup: (a) Schematic of the tank experiments. For sedimentation experiments, spheres were held in place using a neodymium magnet at the drop location. The 80-20 extruded aluminum was used to locate the sphere relative to the front vertical wall, which was moved using the horizontal stage for parallel translation experiments. A camera and tracking software measured the speed of the sedimenting sphere. For torque experiments, vertical and horizontal stages positioned the sphere either near the bottom boundary for perpendicular axis measurements or near the front vertical boundary for parallel axis measurements. Torque was measured with a FUTEK torque sensor and the motor and magnetic encoder were housed inside a 3D-printed structure that was mounted to the active side of sensor. (b) Example showing frequency data (solid blue symbols) read with an Agilent counter at about 1.5 samples per second, and torque data (solid black curve) were read using a DAQ at about 10 samples per second. Torque per frequency (solid red curve) was calculated by interpolating frequency data to the time where torque was measured so that dimensionless torque could be calculated. (c) Dimensionless drag for sphere motion parallel to the boundary, $F_\parallel$, versus scaled boundary distance, d/R, where R is the radius of the sphere and d is the distance from the center of the sphere to the boundary. The solid black curve shows the theory by Lee and Leal [14]. $F_\parallel$ is scaled using μ, the dynamic viscosity, the speed v, and R. Blue triangles are the raw tracking data and the green circles are the same data with the additional drag from the other five boundaries removed using Lee and Leal's theory. The raw data show a systematic difference from the theory at all boundary distances, whereas the green circles agree well once the finite size effects are removed.

of the tank. These boundary effects were clearly evident in our data because the speed changed as the sphere fell, whereas the theory predicts a constant drag at a given distance to the vertical boundary. We used the theory to predict the additional drag at each depth from the upper and lower boundaries and subtracted it from the experimental measurement. The sphere was also moving parallel to three other vertical walls, which was calculated and subtracted from the data. We then compared the experimental measurement to the theory for a sphere moving parallel to a single infinite boundary. Our method assumed that the tank walls were large enough for the theory to be accurate and that the effects of multiple boundaries added linearly, and it seemed to work very well. For example, Fig. 1(c) shows how the raw data for parallel translation agrees much better with Lee and Leal's theory once the finite size effects are removed, whereas it would not otherwise agree within uncertainty with the theory.

A. Perpendicular and parallel translation

Fluid drag was measured by sedimenting spheres with radius $R = 12.70 \pm 0.03$ mm either perpendicular toward a boundary or parallel to a boundary. The boundary was positioned relative to the spheres using a stage, and the spheres released below the free surface using a magnet (see Fig. 1).

We recorded motion tracking videos of falling spheres at approximately four frames per second using a camera and a 25-mm focal length lens [26] positioned to minimize spherical aberration where the tracking required the most precision.

The fluid drag was calculated as the difference between the buoyant and gravitational forces. The uncertainty in the gravitational and buoyant forces were small and considered negligible. The fluid drag in the z direction, $F_\perp$ and $F_\parallel$, was made dimensionless by dividing by $\mu R v$, where μ is dynamic viscosity of the fluid, R is the radius of the sphere, and v is the instantaneous speed of the sphere. The speed was calculated using a central difference between the tracked positions divided by the time between frames.

Lee and Leal's theory predicts an additional drag of about 10% from the other five boundaries (four vertical and the upper free surface) in the location where drag measurements were made, which was subtracted from the data. We present three trials in Fig. 7(a) showing very little scatter in the perpendicular drag data, so no uncertainty was assigned for these data. When measuring the parallel drag, we tracked the spheres for $\Delta z \approx 70$ mm after they were about 0.10 m below the surface, where the additional drag from the upper and lower boundaries was approximately constant. The uncertainty in the speed was typically $\Delta v \approx 0.3$ mm/s based on uncertainty in position of the sphere, and the propagated error was plotted as vertical error bars.

B. Perpendicular and parallel rotation

Torque measurements used a Bal-Tec® ball gauge: a stainless steel sphere of radius $R = 12.70 \pm 0.01$ mm with a precision hole drilled along a diameter. We removed the manufacturer's supplied shaft and attached a carbon steel rod with radius $r = 4.763 \pm 0.013$ mm and length $L = 304.8 \pm 0.1$ mm. The rod was attached to a DC motor with a magnetic encoder [27], which was mounted inside of a 3D-printed enclosure. The shaft passed through a sleeve bearing, which reduced the sphere's precession to <0.25 mm.

We measured torque and rotation rate for 60–80 rotation periods in both the clockwise and counterclockwise rotation directions at each boundary location, with the difference in the two values used as the uncertainty. The DC motor was under computer control [28] and the motor frequency read using a counter [29] that removed high-frequency noise. A reaction torque sensor [30] measured the torque, and an amplifier [31] amplified the signal, which was fed into a data acquisition system [32] and recorded using matlab software. The frequency and torque data were read at different rates, so we interpolated frequency to the same time points that the torque was read [see Fig. 1(b)]. This interpolation was necessary so that the torque could be expressed as a dimensionless quantity for comparison to theory and simulations.

For perpendicular axis torque measurements, the rod-and-sphere assembly was in the middle of the tank, as shown in Fig. 1. The distance to the bottom boundary was adjusted using a computer-controlled vertical stage with a resolution ± 0.01 mm. For parallel axis measurements, the sphere was at a depth of $0.21z$ m and the distance to the front boundary was adjusted using computer-controlled horizontal stage with resolution ± 0.01 mm.

We used the theory of Jeffrey and Onishi [3,33] to calculate the torque on the rod and subtracted it from the total to isolate the torque of the sphere. The work of Shindell *et al.* [3] verified the accuracy of this theory for torque on a cylinder, but we made preliminary measurements to ensure that we could reproduce that work before using the theory to remove the cylinder torque from the signal. Those preliminary measurements showed percent differences between experiments and theory that were generally $\leqslant 1\%$.

Torque data were made dimensionless using $\mu \Omega R^3$, where μ is the dynamic viscosity, Ω is the angular speed of the rotating sphere, and R is the sphere's radius. Plots of dimensionless torque versus scaled boundary distance are shown in Figs. 7(c) and 7(d) along with the theory and simulations. Lee and Leal's theory predicts negligible contributions to the sphere's torque from finite size effects ($< 0.1\%$), which were ignored.

II. NUMERICAL METHODS

Our primary goal is to develop an optimized method for modeling spheres moving near a boundary using experiments and Lee and Leal's theory [14] to provide reference values for the optimization. We sought to minimize the two primary sources of error in any fluid simulation of a solid object: those from approximations in the method and those from discretizing the object. We first used the MRS developed by Cortez *et al.* [2] to find optimal values for the regularization parameter, ϵ, for a given discretization in free space and then used the GSIRS to simulate the four fundamental motions of a sphere near a surface for comparison with the theory and experiments.

The MRS uses a regularization function $\omega_\epsilon(r)$ to approximate singular point forces arranged on a surface that represents a solid object in Stokes flow. Replacing the singular forcing with a regularization function creates a linear system of equations that can be solved without using specialized methods to accommodate the singularities. The GSIRS extends the MRS to include a solid boundary by imposing a counter Stokeslet, a potential dipole, a Stokeslet doublet, and two rotlets at the image point of each discrete point $\mathbf{x}_k$. The image system thereby imposes a no-slip condition by canceling the fluid velocity at the boundary, as shown in [1].

The GSIRS simulations solve the incompressible Stokes equations with external forcing:

$$\mu\Delta\mathbf{u}(\mathbf{x}) - \nabla p(\mathbf{x}) = -\mathbf{F}(\mathbf{x}) \text{ and}$$
$$\nabla\cdot\mathbf{u}(\mathbf{x}) = 0, \tag{1}$$

where $\mathbf{u}$ is the fluid velocity, p is the pressure, and μ is the dynamic viscosity. The vector field $\mathbf{F}(\mathbf{x})$ is a force density (force per volume) applied to the surface of the sphere.

The discretization of the sphere surface. In this work, we analyze two types of discretization: the 6-patch and the SCVT discretization schemes [see Fig. 2(b)]. The 6-patch method distributes points from a bounding cube to the sphere's surface [2,34–36]. The number of points, N, is determined using the formula $N = 6n^2 - 12n + 8$, where n is the number of points on each edge of the cube. The 6-patch method is commonly used and is easy to implement, but the point distribution is not uniform on the sphere's surface, which may cause errors in the simulation.

Our SCVT simulations used the method of Du *et al.* [22] to produce a nearly uniform point distribution on the sphere using the SCVT software available at Lili Ju's website [37] with a constant density function. Figure 2(b) compares the uniformity of the point distributions for the SCVT and 6-patch methods.

Regularized Stokeslets. Once the sphere is discretized, the forcing $\mathbf{F}$ in Eq. (1) is the sum of the force density applied at the discrete points $\mathbf{x}_k$, $k = 1,\ldots,N$, on the sphere model, defined by $\mathbf{F}(\mathbf{x}) = \sum_{k=1}^{N}\mathbf{f}_k\omega_\epsilon(\mathbf{x}-\mathbf{x}_k)$ where $\mathbf{f}_k$ is a point force at $\mathbf{x}_k$ and $\omega_\epsilon(\mathbf{x})$ is a regularization function having an integral equal to unity over $\mathbf{R}^3$. The MRS provides the exact solution of Stokes equations for this forcing in open space. To enforce the zero-flow boundary condition at a nearby plane, the GSIRS adds the fluid velocity due to several other regularized elements at the image point of $\mathbf{x}_k$, as described above.

The choice of regularization function. In our work, we choose $\omega_\epsilon(r)$ as either

$$\phi_\epsilon(r) = \frac{15\epsilon^4}{8\pi(r^2+\epsilon^2)^{7/2}}$$
$$\text{or}\qquad \psi_\epsilon(r) = \frac{15\epsilon^4(40\epsilon^6 - 132\epsilon^4r^2 + 57\epsilon^2r^4 - 2r^6)}{16\pi(r^2+\epsilon^2)^{13/2}},$$

where $r = |\mathbf{x}-\mathbf{x}_k|$ and ϵ represents the regularization parameter. Two-dimensional plots of $\phi_\epsilon(r)$ and $\psi_\epsilon(r)$ are shown in Fig. 2(a), for reference. We now provide the reasoning for this choice.

For a point force $\mathbf{f}$ applied at $\mathbf{y}$, the regularized Stokeslet velocity, $S_\epsilon(\mathbf{x},\mathbf{y})$, derived using the MRS with an arbitrary regularization function $\omega_\epsilon(r)$, can be expressed as follows:

$$S_\epsilon(\mathbf{x},\mathbf{y})\mathbf{f} = \frac{\mathbf{f}}{8\pi r}H_1(r) + \frac{(\mathbf{f}\cdot(\mathbf{x}-\mathbf{y}))(\mathbf{x}-\mathbf{y})}{8\pi r^3}H_2(r),$$

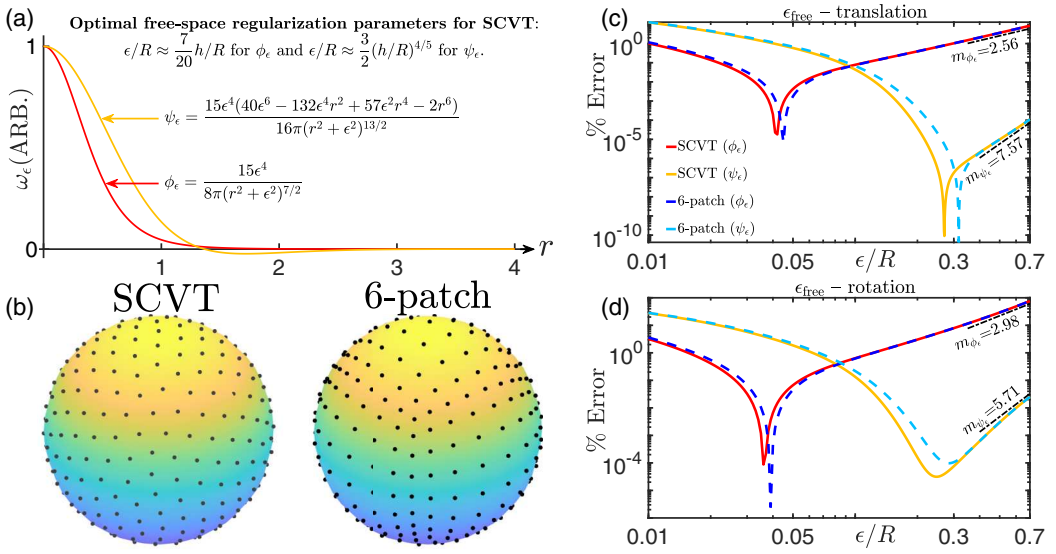

FIG. 2. Finding optimal regularization parameters in free space: (a) Regularization functions ϕ_ϵ and ψ_ϵ are normalized such that their values at the center are equal. (b) Surface discretizations ($N = 296$) on a sphere using SCVT and 6-patch method showing the point distributions. Optimal regularization parameters for (c) translation and (d) rotation in free space with $N = 1000$ (SCVT) and $N = 1016$ (6-patch): the minima of the percent errors for the four different combinations are used to identify the value of the optimal regularization parameters. The decay properties of the regularization functions ϕ_ϵ and ψ_ϵ are shown as black dash-dotted lines that represent the slope of the best fit line used to determine the order m of the regularization error as the regularization parameter, ϵ, is large compared to the average discetization length, h.

where the functions $H_1(r)$ and $H_2(r)$ contain the regularization and approach the value 1 as $r \to \infty$. The singular Stokeslet velocity $S_0(\mathbf{x}, \mathbf{y})$, corresponding to the case without regularization, is obtained by setting $H_1(r) = H_2(r) \equiv 1$ in the expression.

The functions corresponding to $\phi_\epsilon(r)$ are

$$H_1(r) = \frac{r(r^2+2\epsilon^2)}{(r^2+\epsilon^2)^{3/2}} \quad \text{and} \quad H_2(r) = \frac{r^3}{(r^2+\epsilon^2)^{3/2}}$$

and the functions corresponding to $\psi_\epsilon(r)$ are

$$H_1(r) = \frac{r(2r^8 + 9r^6\epsilon^2 + 18r^4\epsilon^4 - 8r^2\epsilon^6 + 16\epsilon^8)}{2(r^2+\epsilon^2)^{9/2}}$$

$$\text{and} \quad H_2(r) = \frac{r^3(2r^6 + 9r^4\epsilon^2 + 12r^2\epsilon^4 + 40\epsilon^6)}{2(r^2+\epsilon^2)^{9/2}}.$$

These functions are shown in Fig. 3.

The error in the computation can be decomposed into two terms: (1) the regularization error, which depends only on the regularization function and the parameter ϵ, and (2) the discretization error, which depends on the same parameters as the regularization error, as well as the average discretization length, $h = \sqrt{\frac{4\pi R^2}{N}}$, where N is the number of discretization points and R is the sphere radius. The regularization error dominates when ϵ is large compared to h. Consequently, even if ϵ is small, provided that the ratio ϵ/h remains large, the overall error is $\mathcal{O}(\epsilon^m)$.

To understand how our choice of a regularization function affects the errors, we consider the convergence of the regularized Stokeslet to the singular one in the far field. We base this discussion

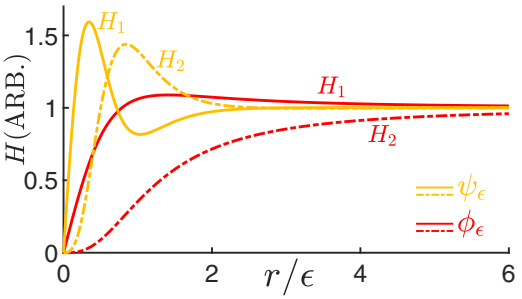

FIG. 3. Regularization functions $H_1(r)$ and $H_2(r)$ for both regularizing functions $\phi_\epsilon(r)$ and $\psi_\epsilon(r)$. These functions all approach unity as $r/\epsilon \to \infty$, but ψ_ϵ satisfies higher moments, so $H_1(r)$ and $H_2(r)$ approach one more rapidly.

on analytical results presented in [38,39] that show that the regularization error can be reduced by imposing moment conditions on $\omega_\epsilon(r)$. Specifically, the second-moment condition

$$M_2 = \int_0^\infty s^4 \omega_\epsilon(s) ds = 0$$

reduces the theoretical regularization error in the far field, although it does not necessarily affect the discretization error. These moment conditions are present in both $H_1(r)$ and $H_2(r)$. As $r \to \infty$, the functions $H_1(r)$ and $H_2(r)$ converge to one at different rates, depending on the chosen regularization function. In fact, assuming $\omega_\epsilon(r)$ has integral equal to 1, the functions derived from $\omega_\epsilon(r)$ satisfy

$$H_1(r) - 1 = \frac{M_2}{3r^2} - \frac{4\pi}{3r^2} \int_r^\infty (3r^2 s^2 - 4r^3 s + s^4) \omega_\epsilon(s) ds,$$
$$H_2(r) - 1 = -\frac{M_2}{r^2} - \frac{4\pi}{r^2} \int_r^\infty (r^2 s^2 - s^4) \omega_\epsilon(s) ds,$$

which helps show the rate of convergence to the singular solution as $r \to \infty$. Note that when r is sufficiently large compared to ϵ, the integration in the last terms is over the far field and the terms containing the integrals decay at a rate related to the decay rate of ω_ϵ, which can be controlled in the construction of the regularization functions. Additionally, when $M_2 \neq 0$, the terms containing M_2 decay as r^{-2} for all regularization functions $\omega_\epsilon(r)$. This implies that the difference $S_\epsilon(\mathbf{x}, \mathbf{y}) - S_0(\mathbf{x}, \mathbf{y})$ approaches zero in the far field at a rate proportional to r^{-2}, regardless of the decay rate of $\omega_\epsilon(r)$. This part of the error can be eliminated by designing a regularization function with zero second moment, $M_2 = 0$.

Furthermore, imposing the additional moment conditions

$$\int_0^\infty s^{2k+1} \omega_\epsilon(s) ds = 0, \quad k \geqslant 1, \tag{2}$$

reduces the theoretical regularization error for r/ϵ larger than a tolerance. This region includes the far field but it can also affect the near field if ϵ is small enough. For example, the regularization function $\psi_\epsilon(r)$ has been designed to have a reduced regularization error for $r/\epsilon \gtrsim 5/2$ (see Fig. 3), which generally corresponds to the far field but can also affect the near field for small enough ϵ.

A closer examination of the two regularization functions in our study reveals the source of their differing decay properties. The regularization function $\omega_\epsilon(r) = \phi_\epsilon(r)$ is positive for all r and integrates to 1, but does not satisfy any additional conditions. In contrast, the regularization function $\omega_\epsilon(r) = \psi_\epsilon(r)$ also integrates to 1 but further satisfies $M_2 = 0$ and the moment condition in Eq. (2)

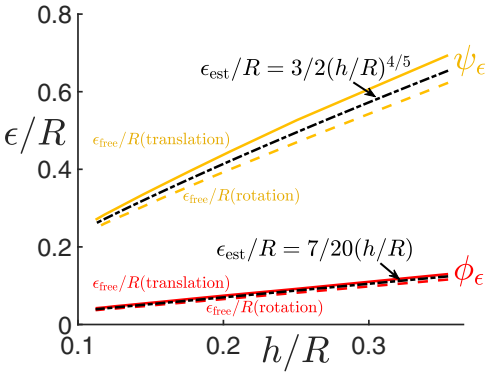

FIG. 4. Optimal regularization parameters in free space for SCVT for different surface discretizations with average discretization lengths, $h = \sqrt{\frac{4\pi R^2}{N}}$, where N is the number of discretization points and R is the sphere radius. The optimal regularization parameter, $\epsilon_{\rm free}/R$, for each regularization function can be estimated as a function of the average SCVT discretization length, h, for both types of sphere motion. The fit function estimates for the regularization parameters are shown as black dash-dotted curves. The relative difference between optimal regularization parameters for rotation and translation is approximately the same for ψ_ϵ and ϕ_ϵ but is exaggerated for ψ_ϵ because of the scale in the plot. The means of the ratio (rotation to translation) of the optimal regularization parameters are 0.91 ± 0.01 for ψ_ϵ and 0.88 ± 0.01 for ϕ_ϵ.

for $k = 1$ and $k = 2$. To meet these conditions, $\psi_\epsilon(r)$ takes on both positive and negative values, which increases the accuracy of the MRS. Moreover, because both functions integrate to one, ψ_ϵ must have a wider positive region to compensate for its negative contribution to the integral. This is reflected in the size of the optimal regularization parameter found in Fig. 4 where we optimized the MRS using the Stokes flow values of drag and torque on a sphere.

Figures 2(c) and 2(d) illustrate the differences in the convergence rates of the regularization errors. Specifically, replacing ϕ_ϵ with ψ_ϵ increases the exponent in $\mathcal{O}(\epsilon^m)$ from $m = 2.56$ to $m = 7.57$ for the translation data and from $m = 2.98$ to $m = 5.71$ for the rotation data, using $N_{\text{6-patch}} = 1016$ and $N_{\rm SCVT} = 1000$ discretization points, respectively. Consequently, ψ_ϵ achieves a higher order of accuracy than ϕ_ϵ, which we attribute to its satisfaction of the additional moment conditions. The simulations in the following sections are designed to assess whether this enhanced accuracy extends to the GSIRS once the images are included. We believe that the increased accuracy observed with ψ_ϵ highlights the significance of the error reduction.

III. RESULTS

A. Comparison of experiments with theory

Figure 7 shows the close agreement between experiments and the theory of Lee and Leal [14], which provides a level of precision unmatched by previous experimental tests of theory. The correspondence relied critically on removing any significant effects of other boundaries. The tank was large enough (approximately $48R \times 48R \times 30R$) that the predicted torque increase was negligible and ignored. The smallness of the predicted effects is consistent with our expectation that the boundary effects on the sphere torque will scale as $1/d^2$, where d is the boundary distance. However, theory predicts a significant increase in the drag on a sedimenting sphere ($\approx$10%) from the other five boundaries. The scaling for increased drag force due to boundary effects we expect to be $1/d$ so the contribution should be much larger for those data, which is what we found, as shown in Fig. 1(c). Other unknown finite size effects such as corners, free surface motion, etc., were small enough that the theory was still accurate in a tank experiment of the size we used.

B. Optimizing the MRS and GSIRS

1. Comparison of optimal regularization parameters in free space

Figures 2(c) and 2(d) allowed us to find optimal regularization parameters for each type of motion (translation or rotation) and each combination of discretization method and regularization function in free space. Given a discretization with N points (and average discretization length, h), the optimal value of ϵ minimizes the percent error relative to theory. The optimal value of ϵ for the SCVT simulations varies smoothly as a function of the average discretization size, h, so we found fit functions that relate ϵ and h for each of the regularization functions (see Fig. 4):

$$\epsilon_{\text{est}}/R \approx \frac{7}{20}h/R \quad \text{for } \phi_\epsilon \qquad \text{and} \qquad \epsilon_{\text{est}}/R \approx \frac{3}{2}(h/R)^{\frac{4}{5}} \quad \text{for } \psi_\epsilon. \tag{3}$$

These formulas allow other researchers to bypass the free-space-optimization process and use the fit functions instead. The 6-patch data have a nonuniform discretization length, so no comparable formulas were calculated. The value of ϵ_{est}/R for ϕ_ϵ is very different than the value found for cylinders by Shindell *et al.* [3]. In their work, they found that $\epsilon/R = (h/R)/6.4$, where h was the discretization length between Stokeslets on a side surface of a cylinder. Thus, we confirm the Shindell *et al.* result that the relationship between h and ϵ depends on the geometry of the solid object being simulated.

2. Comparison of discretization methods

We used the GSIRS, the two regularization functions ϕ_ϵ and ψ_ϵ, and the free-space-optimized regularization parameters found above to simulate drags and torques for the four fundamental motions: perpendicular translation, parallel translation, perpendicular axis rotation, and parallel axis rotation. We used a range of discretization points $100 \leqslant N \leqslant 1016$ and randomly rotated the discretized sphere with respect to the boundary to assess the importance of uniformity in the point distribution. We evaluate the performance of the optimized 6-patch and SCVT simulations near the boundary ($d/R \in \{1.089, 1.128, 3.0\}$) in Fig. 5 because both types perform well far from the boundary. We ran 17 280 calculations in total: 4 motions, 2 regularization functions, 2 discretization methods, 10 values of N, 3 distances, and 36 random rotations of each discretization. The SCVT simulations (solid curves) generally outperform the 6-patch method (dashed curves), especially as the number of discretization points increases. The 6-patch simulations show relative standard deviations (RSDs) as high as 90% when the discretization is randomly rotated, while the SCVT has an RSD of 1–2 % across all types of motion near the boundary.

Generally, the percent error decreases as N increases, as expected, but the rate at which this occurs is different. Simulations using ψ_ϵ (solid gold and dashed cyan curves) show a more rapid decrease in percent error as N increases when compared to matched simulations using ϕ_ϵ (solid red and dashed dark blue curves). The data also show that SCVT simulations using either regularization function (solid gold and solid red curves) return a lower mean percent error than the matched 6-patch simulations (dashed cyan and dashed dark blue curves), especially when $N \approx 1000$.

At the distance $d/R = 1.089$, the sphere's edge is closer to the boundary than the average discretization length for all simulations in the top row of Fig. 5. We discuss the importance of the sphere being closer than the average discretization distance below in Sec. III B 3. The simulations from the combination of SCVT and ψ_ϵ (solid gold) show lower percent error in all panels except Fig. 5(b). Furthermore, Table I lists the range of the percent errors for these simulations when $800 \leqslant N \leqslant 1016$: the performance is especially poor for the translational motion perpendicular to the wall as seen in Fig. 5(c) for all combinations. Near the boundary the percent error is very sensitive to small changes in the regularization parameter ϵ as shown in Sec. III B 4.

When $d/R = 1.128$, as shown in the second row of Fig. 5, the SCVT simulations using ψ_ϵ generally converge more quickly, have lower variance, and achieve lower percent error at the highest value of N, as shown in Figs. 5(e)–5(g), though there is no clearly superior method for simulating the perpendicular torque shown in Fig. 5(h). Here we note that the simulated sphere is farther from

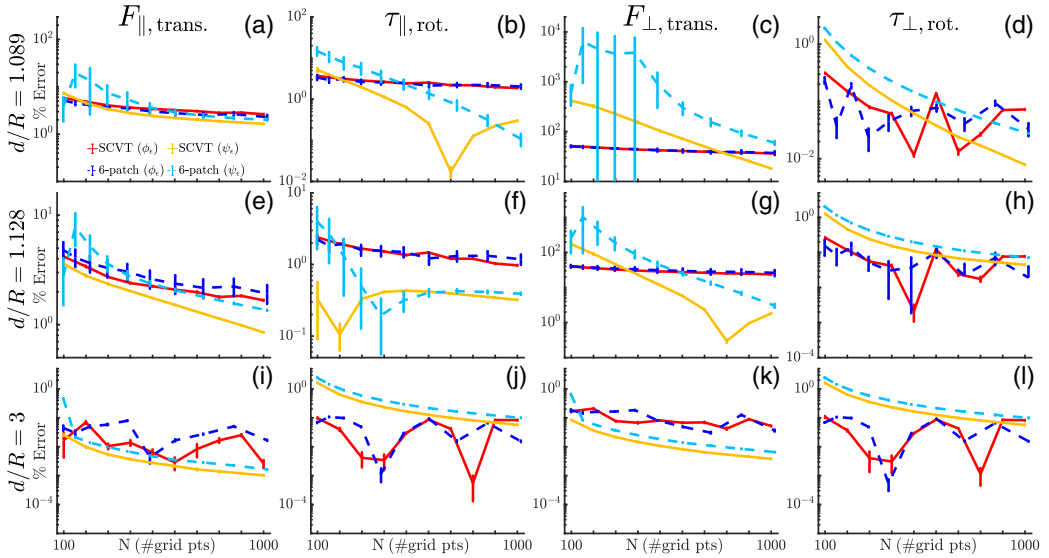

FIG. 5. Percent errors relative to theory for the four fundamental sphere motions versus number of discretization points, N, at three locations near a boundary: $d/R = 1.089$ (top row), $d/R = 1.128$ (middle row), and $d/R = 3$ (bottom row): [(a), (e), (i)] drag when moving parallel to the boundary, [(b), (f), (j)] torque with the rotation axis parallel to the boundary, [(c), (g), (k)] drag when moving perpendicular to the boundary, and [(d), (h), (l)] torque with the rotation axis perpendicular to the boundary. Solid lines are SCVT simulations and dashed lines show 6-patch data. Red and dark blue curves are simulations using the ϕ_ϵ regularization function, whereas the cyan and gold curves used the new ψ_ϵ regularization function. Each motion was simulated 36 times with the discretization rotated along a random axis and a random angle. Percent error is the mean of the 36 random trials, and error bars are the standard deviation. The SCVT simulations show a lower variance when compared with the 6-patch simulations (see Fig. 6), and the SCVT generally shows smaller percent errors at the highest values of N. The top row shows that once the gap between the edge of the sphere and the boundary, $d/R - 1 = 0.089$, is smaller than a discretization length, $h/R \in [0.11, 0.35]$ for $98 \leqslant N \leqslant 1016$, the percent error and the variance are higher.

the boundary than the average discretization length once $N > 800$. Either the ψ_ϵ percent error is lower or it is very low for all simulation types. Thus, we consider ψ_ϵ a better regularization function whenever the sphere is sufficiently resolved in the simulations.

The last row of Fig. 5 when $d/R = 3$ shows that the percent error is less sensitive to changes in the regularization function and discretization method when the sphere is far from the boundary. The 6-patch and SCVT show similar performance in mean percent error values, which are small for all simulation types.

TABLE I. Range of percent errors for $d/R = 1.089$ and $800 \leqslant N \leqslant 1016$ for four motions. The range was determined by finding the smallest (min) and largest (max) percent errors of each combination of the discretization method (SCVT or 6-patch) and the regularization function (ϕ_ϵ or ψ_ϵ) for all these values of N.

Min–max	SCVT (ϕ_ϵ)	SCVT (ψ_ϵ)	6-patch (ϕ_ϵ)	6-patch (ψ_ϵ)
$F_{\parallel,\text{trans}}$	2.604–2.911	1.643–1.888	1.635–3.940	1.837–3.256
$\tau_{\parallel,\text{rot}}$	1.837–2.213	0.113–0.307	1.076–2.818	0.004–1.453
$F_{\perp,\text{trans}}$	36.167–38.847	17.761–36.160	24.117–46.052	35.575–231.592
$\tau_{\perp,\text{rot}}$	0.023–0.076	0.007–0.018	0.005–0.137	0.0269–0.064

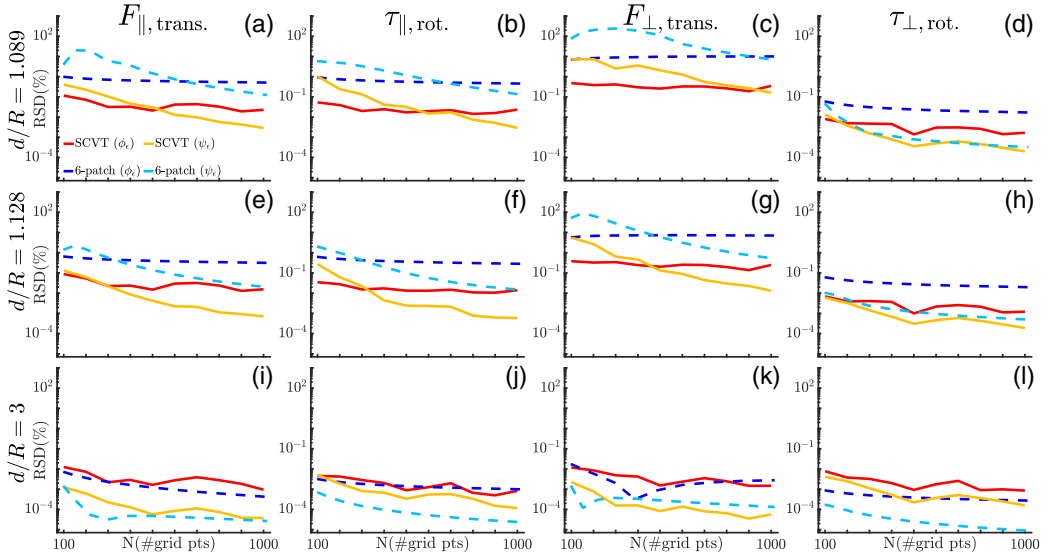

FIG. 6. The plots show the relative standard deviation (RSD) as a percentage versus number of discretization points, N, at $d/R \in \{1.089, 1.128, 3.0\}$. [(a), (e), (i)] Drag for motion parallel to the boundary, [(b), (f), (j)] torque for rotation with axis parallel to the boundary, [(c), (g), (k)] drag for motion perpendicular to the boundary, and [(d), (h), (l)] torque for rotation with axis perpendicular to the boundary. Red and dark blue curves are simulations that used the ϕ_ϵ regularization function, whereas the gold and cyan curves used the ψ_ϵ regularization function. Each sphere motion was simulated 36 times with the sphere's discretization rotated randomly. The SCVT simulations show a lower RSD when compared with the 6-patch simulations for most motion types especially when the sphere is close to the boundary. The SCVT simulations using the ψ_ϵ regularization function generally show the lowest RSD as N becomes large. Away from the boundary, all simulation types in panels (i)–(l) show a low percent error relative to the theory.

We extended our analysis of the variance in the different simulation types by plotting the RSD of the mean value of force and torque in Fig. 6. The plots of RSD versus number of discretization points, N, show that the SCVT simulations are much less sensitive to the orientation of the sphere with respect to the boundary at any boundary distance. The 6-patch simulations in Fig. 6(c) have an RSD as high as 90% for the perpendicular translation data while the SCVT has an RSD of 1–2 %. The asymmetry of the 6-patch discretization method therefore requires care to ensure the discretization does not skew results near the boundary and requires averaging of different orientations for accuracy, whereas the SCVT shows very little variance in the computed force or torque. Thus, the advantage of using a uniform discretization method is that averaging over different orientations, as we have done for analysis, is not necessary.

3. Comparison of the optimized simulations with experiment and theory

We used the higher-order regularization function, ψ_ϵ, the SCVT discretization method ($N = 1000$), and the free-space-optimized value for the regularization parameter to simulate sphere motion for a large range of boundary distances for comparison with the experiments and theory, as shown in Fig. 7.

The perpendicular translation, $F_\perp$, data in Fig. 7(a) shows that the data all agree for most boundary distances, except very near the boundary, which is consistent with what we found in Fig. 5. Near the boundary, the dimensionless drag increases by a factor of nearly 10^4, and the distance moved between frames in the experiment is around 10 μm, which highlights the precision needed in the motion tracking. However, the inset in Fig. 7(a) shows that the numerical simulations

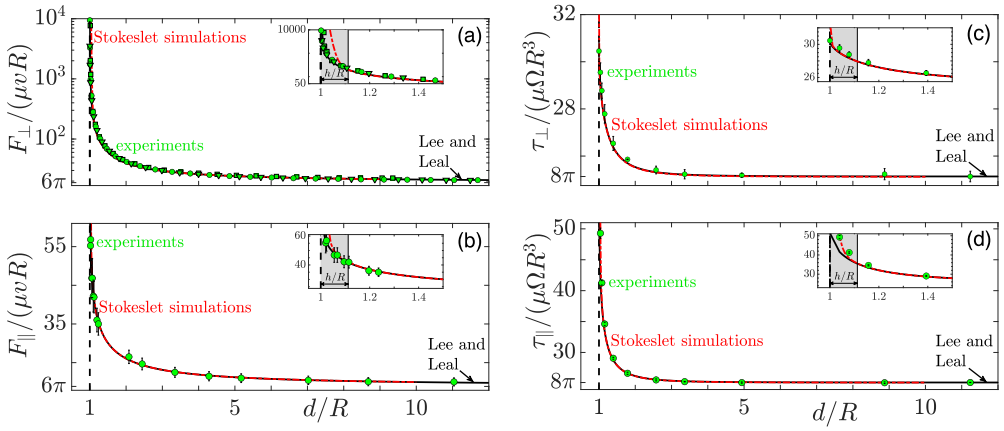

FIG. 7. Dimensionless drag and torque versus scaled boundary distance, d/R, on a sphere, where R is the radius of the sphere, and d is the distance from the center of the sphere to the boundary. The solid black curves show the theory by Lee and Leal [14], the solid red curves are optimized GSIRS simulations, and the solid green symbols are dynamically similar experiments. (a) Translation perpendicular to a boundary, $F_\perp/(\mu v R)$, and (b) motion parallel to a boundary, $F_\parallel/(\mu v R)$, scaled using μ, the dynamic viscosity, v is the speed, and R. [(c), (d)] Dimensionless torque versus boundary distance for rotation with the axis perpendicular to the surface, $\tau_\perp/(\mu\Omega R^3)$, and with the rotation axis parallel to the boundary, $\tau_\parallel/(\mu\Omega R^3)$. Experiments, simulations, and theory all agree very well outside of the near-boundary region. The experiments agree with theory at most distances except for $\tau_\parallel$, which diverges from theory near the boundary. We attribute this difference to locating the edge of the sphere (see text). However, the insets show the simulations diverge from theory in the near-boundary region for three of the four motions, which occurs when the edge of the sphere is closer than the average discretization size ($h/R = 0.11$) to the boundary. The inset in panel (c) shows that the simulations perform better for $\tau_\perp$ than the other three motions.

diverge systematically from the theory and experiments when the sphere is near $d/R \approx 1.1$. We find the same systematic divergence from theory when using either ϕ_ϵ or ψ_ϵ and find a similar result for other motions, as described below.

Figure 7(b) shows that the parallel translation, $F_\parallel$, simulations perform better overall than the simulations of perpendicular translation, but similarly, the inset shows deviation from theory in the simulations at about $d/R = 1.1$. The maximum drag increase expected here is a factor of about 10 compared to about 10^4 in the perpendicular translation case, which may explain why the deviation is less pronounced until about $d/R = 1.05$.

Figure 7(c) shows that the three data types—theory, experiments, and simulations—all agree very well for torque with the rotation axis perpendicular to the boundary, $\tau_\perp$. The deviation of the simulations from experiments and theory occurs much closer to the boundary and is smaller. However, $\tau_\perp$ shows the least increase near the boundary among the four motions studied.

Torque with the rotation axis parallel to the boundary, $\tau_\parallel$, is shown in Fig. 7(d). The experiments show some systematic deviation from theory, which we attribute to uncertainty in establishing when $d/R = 0$ because the precession of the sphere (0.25 mm) is the same order as the closest boundary distance ($d = 0.5$ mm). The simulations also show systematic deviation from theory at about $d/R = 1.1$ in these data.

In summary, none of the simulation types perform well when the sphere is located very near the boundary (see insets in Fig. 7). Other researchers such as Zheng *et al.* [40] have suggested that the method of images does not perform well when the gap between an object and the wall is smaller than the size of the regularization parameter ϵ. In our data, the normalized gap where numerical simulations diverge from theory ($h/R = 0.11$) is larger than the optimal value of $\epsilon/R = 0.042$ for ϕ_ϵ and smaller than the optimal value of $\epsilon/R = 0.27$ for ψ_ϵ. Instead, we find that the

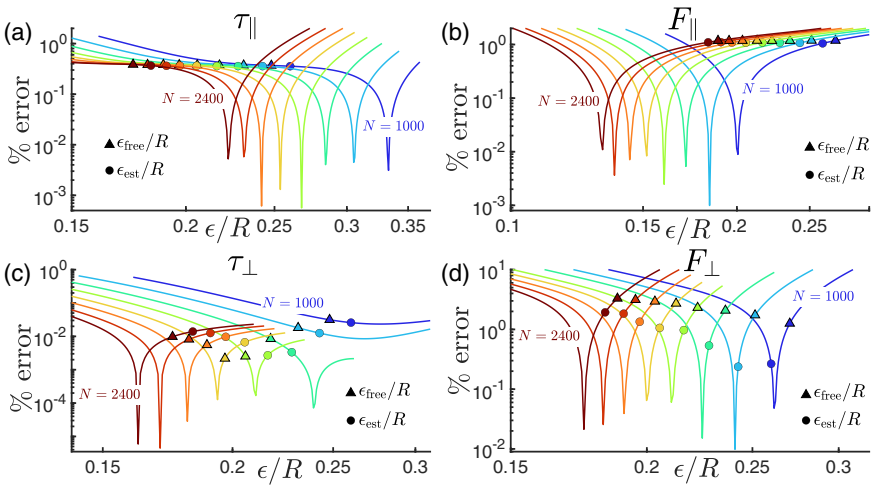

FIG. 8. Optimizing the regularization parameter ϵ when the edge of the sphere is one discretization length h from the boundary. The simulations use the SCVT discretization method and the ψ_ϵ regularization function for eight values of N from 1000 to 2400 in increments of 200 corresponding to eight different discretization lengths h. The solid curves are the percent error as the regularization parameter is varied for each value of N, which all show a clear minimum that could be used to minimize the difference between simulations and theory (see Fig. 9). By contrast, solid triangles are the free-space-optimized values of ϵ_{free}/R found using the MRS simulations, as described in Fig. 4. The solid circles are the estimated values of $\epsilon_{\text{est}}/R = 3/2(h/R)^{4/5}$ using the fit function as described in Fig. 4. Using either the free-space-optimized or the estimated value of the regularization parameter introduces a small percent difference relative to theory near the boundary, but slightly better performance farther from the boundary.

GSIRS simulations diverge from theory when the gap size is within the average discretization size. Section III B 4 discusses options for minimizing the error very near the boundary, but our result gives researchers an empirical rule to identify boundary distances where the GSIRS may not be accurate, i.e., when $d/R < 1 + h/R$.

4. Minimizing error very near the boundary

Our approach to optimizing the simulations far from the boundary by using MRS calculations, as described in Sec. II, results in small percent errors for all boundary distances except when the sphere's edge is closer to the boundary than a discretization length (insets of Fig. 7). We investigated this limitation of our optimization method by varying N from 1000 to 2400 in increments of 200 and placing the sphere at $d = h + R$ to the wall. We simulated all four motions near the boundary using the ψ_ϵ regularization function for a range of ϵ values to find the optimal regularization parameter at this distance, as shown in Fig. 8. The near-boundary optimal regularization parameter at the distance $d = h + R$ is the one that minimizes the percent error between the simulations and the theory.

The presence of the boundary breaks the symmetry of motion in free space and results in four different optimal regularization parameters rather than two very similar values of ϵ_{free}, as found for translation and rotation (see Fig. 4). In free space, the values of ϵ_{free} for translation (0.270) and rotation (0.248) differ by about 9% for ψ_ϵ with SCVT discretization of $N = 1000$, whereas near the boundary the optimal values of ϵ vary by more than 50% for the different motions. Figure 8 also shows that the varying N changes the size of the optimal regularization parameter because the minima occur at different locations h. Thus, not only does the average discretization size h provide the threshold for the minimum gap between the edge of the sphere and the wall, but also the optimal regularization parameter depends on h near the boundary.

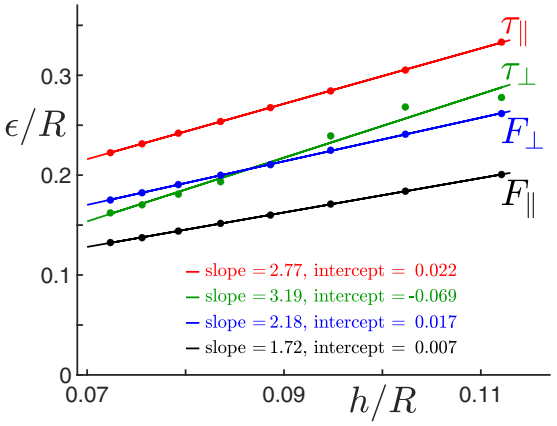

FIG. 9. Optimal regularization parameter ϵ plotted as a function of discretization length h. In these data, the edge of the sphere is located at a distance h from the boundary. Optimal regularization parameters are found for each of the four motion types by identifying the minima of the curves shown in Fig. 8. The solid lines are linear fits to the data with coefficients shown. The data are fit very well except in the case of perpendicular torque $\tau_\perp$, which shows some scatter from the linear fit for the three lowest resolutions: $N = 1000$, 1200, and 1400. The data show that there is no single optimal choice for the regularization parameter once the sphere is near the boundary, whereas in free space, the optimal value of the regularization parameter is more similar for translation and rotation, as shown in Fig. 4.

Assuming h is small, i.e., the sphere is moving near the boundary, we can use Taylor series to expand the regularization parameter for each fundamental motion to first order,

$$\epsilon(h) \approx \epsilon(0) + \left.\frac{d\epsilon}{dh}\right|_{h=0} h \tag{4}$$

Figure 9 shows the results of optimizing ϵ at a boundary distance equal to the discretization length h over a range of values and fitting Eq. (4) to the data. Near the boundary, small changes in ϵ lead to large changes in the percent error, while far from the boundary, the percent error is less sensitive to changes in ϵ (Fig. 2).

Therefore, when a particular simulation requires some motion to be near a boundary, an alternative approach to reduce error is to determine the closest distance desired *a priori* and then to optimize the regularization parameter at that distance; motion farther from the surface using the same regularization parameter value will also have small errors. For the motion of a sphere, the fit values shown in Fig. 9 may be used to determine appropriate regularization parameters, though each component of motion would have to be simulated separately with a different value of ϵ. This requirement makes our free-space-optimization method a better choice in most situations.

IV. DISCUSSION

The goal of this work is to develop methods to precisely calibrate the method of images for regularized Stokeslet simulations of spheres moving near a boundary. We used measurements of forces and torques from macroscopic experiments to verify the theory by Lee and Leal [14] before using it for optimization. We chose this theory because it includes all four motions even though other analyses give the same predictions (see, for example [11–13]). Our results provide a comprehensive experimental test of these theories, and Fig. 7 also shows that the theory can be used effectively to remove the finite size effects in an experiment. Considering the significant finite size effects ($\approx$10%) and that the tank has corners, etc., the theory worked very well despite its underlying assumptions of zero Reynolds number and an infinite plane boundary.

Using the verified theory, we developed computational methods to accurately model a sphere moving near a boundary. Our method of finding the optimal regularization parameter, ϵ_{free}, provides a computationally efficient way to optimize the GSIRS, whether using the ϕ_ϵ or the ψ_ϵ regularization function, though we find the latter to be a better choice because of its higher order of accuracy for the regularization error, as shown in Figs. 2 and 5. The fit functions in Eq. (3) allow researchers to bypass the optimization process and set a value of ϵ when using either ϕ_ϵ or ψ_ϵ.

Our results also confirm the finding of Shindell *et al.* [3]: the relationship between discretization size and regularization parameter depends on the shape of the simulated object. Thus, using a value for the regularization parameter that is not optimized for a given geometry introduces unknown errors into MRS and GSIRS simulations. However, researchers have three geometries for which the regularization parameter has been optimized, which may help better estimate an appropriate value for other geometries.

We also show that the SCVT discretization method is superior to the 6-patch method. The asymmetry of the 6-patch point distribution causes significantly increased variance in the computed values when the sphere is rotated with respect to the boundary (see Figs. 5 and 6). Thus, a discretization method that creates a more uniform point distribution should be chosen when simulating a sphere moving near a boundary. The uniformity helps to reduce the numerical errors and variance in the computed force or torque. Any mesh generator that can produce approximately equally spaced points on a sphere (for example, a Fibonacci lattice) should produce results similar to those from SCVT.

We found that the GSIRS lost accuracy when the sphere was near the boundary (see the insets in Fig. 7). Zheng *et al.* [40] commented that the image system may suffer inaccuracy within distances of order of the regularization parameter from nonzero values of the image sources crossing the boundary. Our GSIRS simulations using ϕ_ϵ, the same regularization function as used by Zheng *et al.*, show high percent errors from theory when the distance of the edge of the sphere ($0.089R$) was larger than the size of the regularization parameter ($\epsilon = 0.042R$) (see the top row of Fig. 5). Furthermore, we found large differences relative to theory at the same distance when using ψ_ϵ, and, in this case, the gap is much smaller than the size of the regularization parameter, $\epsilon = 0.27R$. The force distribution is modified near the wall due to the presence of the image system, but we find that a good empirical rule for establishing the number of discretization points needed is to determine the closest distance to the boundary that will be simulated and ensure that the average discretization length is smaller than this distance. Our method of optimizing the simulations in free space and this discretization rule will give better results than using the size of the regularization parameter to establish the limits of accuracy. These increased errors near the boundary are likely important for other geometries, including simulations of compound objects that approach a boundary, which implies increased care is needed whenever the GSIRS is used for very-near-boundary simulations.

In conclusion, using the GSIRS with a uniform point distribution such as the SCVT, a high-order regularization parameter function such as ψ_ϵ, and a free-space-optimized regularization parameter provides an efficient and accurate method for simulating spheres moving near a boundary providing the edge of the sphere is kept outside of the average discretization length. These results along with our experimental validation of theory and matlab and python implementations of Lee and Leal's theory, MRS, and GSIRS provide important resources for other researchers using the GSIRS or other simulation methods to assess the accuracy of their numerical results. Evaluation of our optimization strategy and calibration process for other geometries including compound objects such as a sphere with a helical flagellum is warranted. Future study might also explore the geometric dependence of the regularization parameter on local curvature and other factors to provide general guidance in its selection.

ACKNOWLEDGMENTS

This research was supported by the collaborative NSF grant through the Physics of Living Systems (Grant No. PHY-2210609 to H.N., A.G., O.S., and F.H., and Grant No. PHY-2210610 to B.R.,

K.B., and J.M.) and the collaborative NSF grant through the DMS/NIH-NIGMS Initiative to Support Research at the Interface of the Biological and Mathematical Sciences (Grant No. DMS-2054333 to R.C. and Grant No. DMS-2054259 to H.N. and A.G.). We thank Trinity University for the provision of computational resources and NSF MRI-ACI-1531594 for providing Trinity the High Performance Scientific Computing Cluster. We thank the Centre College Faculty Development Fund and Gary Crase for assistance with experimental equipment and design.

B.R., H.N., R.C., O.S., and F.H. were involved in conceptualization of the project. For the original draft, B.R. wrote, reviewed, and edited the manuscript while H.N. and R.C. wrote the Numerical Methods section. B.R., H.N., R.C., and O.S. worked together on revising the final draft. B.R. designed the experiments and analyzed the results. B.R. and O.S. created the numerical implementation of theory in matlab. B.R. mentored two undergraduate students (K.M.B. and J.M.) to conduct experiments, analyze the results, and assist in developing the additional information including a python version of the theory. H.N. designed the numerical codes and mentored one undergraduate student (A.G.) on model development and implementation to generate simulation results and data analysis. A.G. conducted simulations and analyzed the results. R.C. provided the regularization function ψ_ϵ and insights into the GSIRS. All authors reviewed the manuscript.

The authors declare no competing interests.

DATA AVAILABILITY

The data that support the findings of this article are openly available [41].

APPENDIX A: NUMERICAL SIMULATION DETAILS

We used three sets of simulations to minimize the error relative to theory when simulating the sphere motions we studied in our experiments. The first set of simulations used the MRS to understand which regularization function, ϕ_ϵ or ψ_ϵ, converges most rapidly as the regularization parameter size increases. These simulations were also used to find an optimal value for the regularization parameter ϵ for each combination of the regularization function and discretization method. The second set of simulations used GSIRS and the optimal regularization parameters from the first set of simulations to assess how orientation relative to the boundary of the 6-patch and SCVT discretizations affects the percent error when compared to theory. We then simulated sphere motion for a large range of boundary distances using our free-space-optimized regularization parameters, SCVT, and the high-order regularization function ψ_ϵ for comparison with the experiments and theory. For the last set of simulations, we further explored how to optimize values for the regularization parameter at very small boundary distances for the sphere motions.

We incremented the regularization parameters from $\epsilon/R = 0.01$ to 0.7 in steps of 0.001 for each motion, each type of regularization function, and each discretization method. We varied the regularization parameter for each type of motion and found the value of ϵ that minimized the difference between the simulation and theory. The range of simulation parameters used is reported in Table II. We then computed the percent errors between the simulations and theory versus regularization parameter size for each motion [see, for example, Figs. 2(c) and 2(d)]. Overall, 55 280 calculations were conducted: 2 motion types (translation and rotation), 2 regularization functions (ϕ_ϵ, ψ_ϵ), 2 discretization methods (SCVT, 6-patch), 10 values of N, and 691 values of regularization parameters, ϵ.

APPENDIX B: NUMERICAL IMPLEMENTATION OF LEE AND LEAL THEORY

The bipolar coordinate theory by Lee and Leal [14] gives a series solution to the force and torque on a sphere moving near an infinite boundary and includes a slip coefficient λ that represents the degree of slip at the boundary from no slip, $\lambda = \infty$, to free slip, $\lambda = 0$. We solve their equations listed below using both matlab and python GUI programs for convenience. The matlab

TABLE II. Parameters values used in numerical simulations.

Parameter	Value
Dynamic viscosity	$\mu = 10^{-3}$ Pa s
Sphere radius	$R = 1$ μm
Translation speed	$U = 1$ μm/s
Rotation rate	$\Omega/(2\pi) = 154$ Hz
Discretization no.	N(6-patch) ∈ {98, 152, 218, 296, 386, 488, 602, 728, 866, 1016}
	N(SCVT) ∈ {100:100:2400}

code includes a GUI interface using matlab's AppDesigner functionality tested on R2021 and later. We also provide a python GUI program that contains the same basic features tested on python 3.12.1, which requires the PySimpleGUI, numpy, and pyperclip packages. The inputs to the GUI's are as follows:

(1) Radius of the sphere: The default is unity since all values are scaled using the radius.

(2) Viscosity ratio: The range is $0 > \lambda \leqslant \infty$. The default is ''`inf`'', i.e., infinity, which is equivalent to a no-slip boundary near the sphere. Implementing $\lambda = 0$ was only included for perpendicular torque calculations since Lee and Leal developed an analytic expression. Otherwise, a very small nonzero value should be used in this limit.

(3) Number of terms: This is the truncation value for the infinite sums.

(4) Boundary distances: The range is $1 < l = d/R \leqslant \infty$. These are boundary distances at which the theoretical values will be calculated. Dimensional values will be scaled by the sphere radius. When $l/R = 1$, the theory diverges.

(5) Motion type: There are four radio buttons provided for selecting which of the four fundamental types of motion are to be calculated.

The MATLAB GUI calls one of three primary functions, for which we have also coded python equivalents:

(1) `motion_parallel.m` and `motion_parallel.py` calculate the force and torque for rotation or translation parallel to a plane interface. The inputs are l, λ, N, and `motion`, where `motion=2` is for translation and `motion=4` is for rotation following the notation of Lee and Leal.

(2) `rotation_perpendicular.m` and `rotation_perpendicular.py` calculate the torque for rotation perpendicular to the boundary. The inputs are l, λ, and N. Here all force components are zero.

(3) `translation_perpendicular.m` and `translation_perpendicular.py` calculate the additional drag force for motion perpendicular to a plane interface. The inputs are l, λ, and N. Here all torques are zero.

The outputs from the function are the increase in the vector force and torque components so conversion to dimensional units is done by multiplying all force values by the Stokes drag $6\pi\mu Rv$ and by multiplying all torque values by the torque on a sphere far from a boundary $8\pi\mu R^3\Omega$, where μ is the viscosity of the fluid, R the radius of the sphere, v is the speed of the sphere, and Ω is the angular frequency.

Below is the system of equations developed by Lee and Leal that are solved in our matlab and python codes [14]. We have preserved their numbering for easy reference and included comments in the matlab and python codes that identify how each equation is used.

In their work, translation perpendicular to a plane interface or rotation with axis perpendicular to the interface results in $m = 0$, whereas parallel motion results in $m = 1$, in the equations below.

The equations result in seven unknown coefficients: A_n^m, B_n^m, C_n^m, E_n^m, F_n^m, G_n^m, and H_n^m. They note that $D_n^m = \hat{D}_n^m = \hat{C}_n^m = 0$ because of Eqs. (32) and (37) below. There are seven coefficients for the

other fluid denoted as $\hat{A}_n^m$, $\hat{B}_n^m$, $\hat{D}_n^m$, $\hat{E}_n^m$, $\hat{F}_n^m$, $\hat{G}_n^m$, and $\hat{H}_n^m$.

$$l \equiv d/R \qquad (2)$$

$$\eta_0 = -\cosh^{-1}(l) \text{(see p. 205 of [14])}$$

For the upper fluid and all n and m:

$$\hat{A}_n^m = -\hat{B}_n^m, \quad \hat{C}_n^m = -\hat{D}_n^m, \quad \hat{E}_n^m = -\hat{F}_n^m, \quad \text{and} \quad \hat{G}_n^m = -\hat{H}_n^m. \qquad \text{(B1)}$$

For $m = 0$

$$-\tfrac{1}{2}nA_{n-1}^0 + \tfrac{5}{2}A_n^0 + \tfrac{1}{2}(n+1)A_{n+1}^0 - nD_{n-1}^0 + (2n+1)D_n^0 - (n+1)D_{n+1}^0$$
$$+n(n-1)E_{n-1}^0 - 2n(n+1)E_n^0 + (n+1)(n+2)E_{n+1}^0 = 0$$

and

$$-\tfrac{1}{2}nB_{n-1}^0 + \tfrac{5}{2}B_n^0 + \tfrac{1}{2}(n+1)B_{n+1}^0 - nC_{n-1}^0 + (2n+1)C_n^0 - (n+1)C_{n+1}^0$$
$$+n(n-1)F_{n-1}^0 - 2n(n+1)F_n^0 + (n+1)(n+2)F_{n+1}^0 = 0,$$

while for $m \geqslant 1$,

$$-\tfrac{1}{2}(n-m)A_{n-1}^m + \tfrac{5}{2}A_n^m + \tfrac{1}{2}(n+m+1)A_{n+1}^m - (n-m)D_{n-1}^m + (2n+1)D_n^m$$
$$-(n+m+1)D_{n+1}^m + \tfrac{1}{2}(n-m)(n-m-1)E_{n-1}^m - (n-m)(n+m+1)E_n^m$$
$$+\tfrac{1}{2}(n+m+1)(n+m+2)E_{n+1}^m - \tfrac{1}{2}G_{n-1}^m + G_n^m - \tfrac{1}{2}G_{n+1}^m = 0$$

and

$$-\tfrac{1}{2}(n-m)B_{n-1}^m + \tfrac{5}{2}B_n^m + 1/2(n+m+1)B_{n+1}^m - (n-m)C_{n-1}^m + (2n+1)C_n^m$$
$$-(n+m+1)C_{n+1}^m + \tfrac{1}{2}(n-m)(n-m-1)F_{n-1}^m - (n-m)(n+m+1)F_n^m$$
$$+\tfrac{1}{2}(n+m+1)(n+m+2)F_{n+1}^m - \tfrac{1}{2}H_{n-1}^m + H_n^m - \tfrac{1}{2}H_{n+1}^m = 0$$
$$D_n^m = \hat{D}_n^m = 0 \quad \text{for all} \quad n, m. \qquad \text{(B2)}$$

For all m, η, and ξ, we require

$$-\frac{(n-m-1)}{2n-1}\left(F_{n-1}^m - \hat{F}_{n-1}^m\right) + \left(F_n^m - \hat{F}_n^m\right) - \frac{n+m+2}{2n+3}\left(F_{n+1}^m - \hat{F}_{n+1}^m\right)$$
$$= -\frac{1}{2}\left[\frac{1}{2n-1}\left(B_{n-1}^m - \hat{B}_{n-1}^m\right) - \frac{1}{2n+3}\left(B_{n+1}^m - \hat{B}_{n+1}^m\right)\right],$$
$$\text{for } m = 0: \quad H_n^0 = \hat{H}_n^0. \qquad \text{(B3a)}$$

For $m \geqslant 1$,

$$-\frac{(n-m+1)}{2n-1}\left(H_{n-1}^m - \hat{H}_{n-1}^m\right) + \left(H_n^m - \hat{H}_n^m\right) - \frac{n+m}{2n+3}\left(H_{n+1}^m - \hat{H}_{n+1}^m\right)$$
$$= -\frac{1}{2}\left[-\frac{(n-m)(n-m+1)}{2n-1}\left(B_{n-1}^m - \hat{B}_{n-1}^m\right) + \frac{(n+m)(n+m+1)}{2n+3}\left(B_{n+1}^m - \hat{B}_{n+1}^m\right)\right]. \qquad \text{(B3b)}$$

For $m = 0$,

$$G_n^0 = \lambda \hat{G}_n^0. \qquad \text{(B3c)}$$

For all m,

$$\begin{aligned} &E_n^m \sinh(n+1/2)\eta_0 + F_n^m \cosh(n+1/2)\eta_0 \\ &\quad = X_n^m(\eta_0) - \frac{1}{(2n+3)\sinh\eta_0}\left[-Z_{n+1}^m(\eta_0) + C_{n+1}^m \sinh(n+3/2)\eta_0\right] \\ &\qquad + \frac{1}{(2n-1)\sinh\eta_0}\left[-Z_{n-1}^m(\eta_0) + C_{n-1}^m \sinh(n-1/2)\eta_0\right]. \end{aligned}$$

For $m = 0$,

$$G_n^0 \sinh(n+1/2)\eta_0 + H_n^0 \cosh(n+1/2)\eta_0 = Y_n^0(\eta_0). \tag{B3d}$$

For $m \geqslant 1$,

$$\begin{aligned} &G_n^m \sinh(n+1/2)\eta_0 + H_n^m \cosh(n+1/2)\eta_0 \\ &\quad = Y_n^m(\eta_0) + \frac{(n+m)(n+m+1)}{(2n+3)\sinh\eta_0}\left[-Z_{n+1}^m(\eta_0) + C_{n+1}^m \sinh(n+3/2)\eta_0\right] \\ &\qquad - \frac{(n-m)(n-m+1)}{(2n-1)\sinh\eta_0}\left[-Z_{n-1}^m(\eta_0) + C_{n-1}^m \sinh(n-1/2)\eta_0\right]. \end{aligned} \tag{B3e}$$

For all m,

$$\begin{aligned} A_n^m \sinh(n+1/2)\eta_0 + B_n^m \cosh(n+1/2)\eta_0 = \frac{-2}{\sinh\eta_0}\Bigg[&\frac{(n-m)}{2n-1}\left\{Z_{n-1}^m(\eta_0) - C_{n-1}^m \sinh(n-1/2)\eta_0\right\} \\ &- \cosh\eta_0\left\{Z_n^m(\eta_0) - C_n^m \sinh(n+1/2)\eta_0\right\} + \frac{n+m+1}{2n+3}\left\{Z_{n+1}^m(\eta_0) - C_{n+1}^m \sinh(n+3/2)\eta_0\right\}\Bigg] \end{aligned}$$

and

$$A_n^m = \lambda \hat{A}_n^m, \quad E_n^m = \lambda \hat{E}_n^m \quad \text{and} \quad G_n^m = \lambda \hat{G}_n^m. \tag{B4}$$

The equations are rewritten so that the left-hand side has only coefficients, and all other terms are on the right-hand side. The matrix is constructed using a for loop, where each column is built as shown in Eq. (B5). The first column ($n = m - 1$) and last column ($n = N + 1$) are removed after construction because the sums should range from $\sum_{n=m}^{N}$, where N is a truncation value chosen for a desired accuracy. The result is a square, banded matrix of size $N \times N$. The rows and columns are constructed as

$$\begin{pmatrix} n-1 & n & n+1 & \dots & N+2 \\ A_{(n-1)}^m & \dots & & & \\ & A_n^m & \dots & & \\ & & A_{(n+1)}^m & \dots & \\ B_{(n-1)}^m & \dots & & & \\ & B_n^m & \dots & & \\ & & B_{(n+1)}^m & \dots & \\ \vdots & & & & \end{pmatrix}. \tag{B5}$$

The resulting matrix is inverted and multiplied times the right-hand side to solve for the coefficients. The force and/or torque for the four fundamental motions are calculated using the following equations.

(1) Force for perpendicular translation:

$$F_x = F_y = 0,$$

$$F_z = \frac{-\sqrt{2}}{3}\sinh\eta_0 \sum_n \left[C_n^0 - (n+1/2)\left(A_n^0 - B_n^0\right)\right], \tag{B6}$$

$$\tau_x = \tau_y = \tau_z = 0. \tag{B7}$$

(2) Force and torque for parallel translation:

$$F_y = F_z = 0,$$

$$F_x = \frac{-\sqrt{2}}{6}\sinh\eta_0 \sum_n \left[G_n^1 - H_n^1 + n(n+1)\left(A_n^1 - B_n^1\right)\right], \tag{B8}$$

$$\tau_x = \tau_z = 0,$$

$$\tau_y = \frac{\sinh^2\eta_o}{12\sqrt{2}} \sum_n \left[\left(2 + e^{(2n+1)\eta_0}\right)\left\{n(n+1)\left(-2C_n^1 - A_n^1\coth\eta_0\right) - (2n+1+\coth\eta_0)G_n^1\right\}\right.$$
$$\left. + \left(2 - e^{(2n+1)\eta_0}\right)\left\{n(n+1)B_n^1\coth\eta_0 + (2n+1+\coth\eta_0)H_n^1\right\}\right]. \tag{B9}$$

(3) Torque for perpendicular axis rotation:

$$F_x = F_y = F_z = 0, \tag{B10}$$

$$\tau_x = \tau_y = 0,$$

$$\tau_z = \frac{\sinh^2\eta_0}{\sqrt{2}} \sum_n n(n+1)\left(-G_n^0 + H_n^0\right). \tag{B11}$$

(4) Force and torque for parallel axis rotation:

$$F_x = \frac{-\sqrt{2}}{6}\sinh\eta_0 \sum_n \left[G_n^1 - H_n^1 + n(n+1)\left(A_n^1 - B_n^1\right)\right],$$

$$F_y = F_z = 0, \tag{B12}$$

$$\tau_x = \tau_z = 0,$$

$$\tau_y = -\frac{1}{3} - \frac{\sinh^2\eta_0}{12\sqrt{2}} \sum_n \left[\left(2 + e^{(2n+1)\eta_0}\right)\left\{n(n+1)\left(-2C_n^1 - A_n^1\coth\eta_0\right) - (2n+1+\coth\eta_0)G_n^1\right\}\right.$$
$$\left. + \left(2 - e^{(2n+1)\eta_0}\right)\left\{n(n+1)B_n^1\coth\eta_0 + (2n+1+\coth\eta_0)H_n^1\right\}\right]. \tag{B13}$$

The MATLAB and PYTHON codes can solve the series solutions quickly even when 10^4 terms are included using a computer with 16 GB of RAM. This number of terms is only necessary for very small boundary distances and changes the result in five or six digits, whereas for many cases $N = 100$ is sufficient.

We compared our results from Lee and Leal [14] with the values from our calculations using $N = 500$ terms and found that the maximal difference is in the sixth digit of accuracy and is generally in the seventh or eighth digit. Our code does not implement $\lambda = 0$, so the values used $\lambda = 10^{-8}$. At this level of precision, it is unclear whose calculations are correct, but the results show that the numerical implementation matches the work of Lee and Leal to a high degree of accuracy. We note that they appear to have a mistake in the value for $\lambda = 10$ and $l = 1.6$, which we believe should be $\tau = -5.26852\times10^{-5}$ but is reported as being $\tau = -5.26852\times10^{-3}$. The former value matches

our results very well, as do all of the other values in the table, so we believe the value in the table is incorrect. Additional materials include a matlab script (Lee_and_Leal_tabulated_values.m) that calculates these differences [41].

[1] R. Cortez, L. Fauci, and A. Medovikov, The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming, Phys. Fluids **17**, 031504 (2005).

[2] R. Cortez and D. Varela, A general system of images for regularized Stokeslets and other elements near a plane wall, J. Comput. Phys. **285**, 41 (2015).

[3] O. Shindell, H. Nguyen, N. Coltharp, F. Healy, and B. Rodenborn, Using experimentally calibrated regularized Stokeslets to assess bacterial flagellar motility near a surface, Fluids **6**, 387 (2021).

[4] E. M. Purcell, The efficiency of propulsion by a rotating flagellum, Proc. Natl. Acad. Sci. USA **94**, 11307 (1997).

[5] G. Li and J. X. Tang, Low flagellar motor torque and high swimming efficiency of *Caulobacter crescentus* swarmer cells, Biophys. J. **91**, 2726 (2006).

[6] C. Li, B. Qin, A. Gopinath, P. E. Arratia, B. Thomases, and R. D. Guy, Flagellar swimming in viscoelastic fluids: Role of fluid elastic stress revealed by simulations based on experimental data, J. R. Soc. Interface **14**, 20170289 (2017).

[7] M. D. Manson, P. M. Tedesco, and H. C. Berg, Energetics of flagellar rotation in bacteria, J. Mol. Biol. **138**, 541 (1980).

[8] J. F. Imhoff, H. G. Truper, and N. Pfennig, Rearrangement of the species and genera of the phototrophic "purple nonsulfur bacteria," Int. J. Syst. Evol. Microbiol. **34**, 340 (1984).

[9] L. Turner, L. Ping, M. Neubauer, and H. C. Berg, Visualizing flagella while tracking bacteria, Biophys. J. **111**, 630 (2016).

[10] G. B. Jeffery, On the steady rotation of a solid of revolution in a viscous fluid, Proc. London Math. Soc. **s2-14**, 327 (1915).

[11] H. Brenner, The slow motion of a sphere through a viscous fluid towards a plane surface, Chem. Eng. Sci. **16**, 242 (1961).

[12] M. E. O'Neill, A slow motion of viscous liquid caused by a slowly moving solid sphere, Mathematika **11**, 67 (1964).

[13] W. R. Dean and M. E. O'Neill, A slow motion of viscous liquid caused by the rotation of a solid sphere, Mathematika **10**, 13 (1963).

[14] S. H. Lee and L. G. Leal, Motion of a sphere in the presence of a plane interface. Part 2. An exact solution in bipolar co-ordinates, J. Fluid Mech. **98**, 193 (1980).

[15] K. Małysa and T. G. M. van de Ven, Rotational and translational motion of a sphere parallel to a wall, Int. J. Multiphase Flow **12**, 459 (1986).

[16] E. R. Dufresne, T. M. Squires, M. P. Brenner, and D. G. Grier, Hydrodynamic coupling of two Brownian spheres to a planar surface, Phys. Rev. Lett. **85**, 3317 (2000).

[17] O. Pitois, C. Fritz, L. Pasol, and M. Vignes-Adler, Sedimentation of a sphere in a fluid channel, Phys. Fluids **21**, 103304 (2009).

[18] G. M. Wang, R. Prabhakar, and E. M. Sevick, Hydrodynamic mobility of an optically trapped colloidal particle near fluid-fluid interfaces, Phys. Rev. Lett. **103**, 248303 (2009).

[19] H. T. Wu, F. Romano, and H. C. Kuhlmann, Attractors for the motion of a finite-size particle in a two-sided lid-driven cavity, J. Fluid Mech. **906**, A4 (2021).

[20] T. Maxworthy, An experimental determination of the slow motion of a sphere in a rotating, viscous fluid, J. Fluid Mech. **23**, 373 (1965).

[21] J. G. Kunesh, H. Brenner, M. E. O'Neill, and A. Falade, Torque measurements on a stationary axially positioned sphere partially and fully submerged beneath the free surface of a slowly rotating viscous fluid, J. Fluid Mech. **154**, 29 (1985).
[22] Q. Du, M. D. Gunzburger, and L. Ju, Constrained centroidal Voronoi tessellations for surfaces, SIAM J. Sci. Comput. **24**, 1488 (2003).
[23] R. Sadourny, Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids, Mon. Weather Rev. **100**, 136 (1972).
[24] Clearco® PSF-60,000cSt polydimethylsiloxane.
[25] Thermo Scientific, Haake Viscotester IQ.
[26] Point Grey GS3-U3-41C6M-C camera and a Fujinon 25-mm focal length lens.
[27] Pololu 4846.
[28] ARDUINO MEGA 2560 with an ADAFRUIT Motor Shield v.2.
[29] Agilent 53132A.
[30] FUTEK TFF400, 10 in-oz.
[31] FUTEK amplifier model IAA100.
[32] National Instruments (NI USB-6211) DAQ.
[33] D. J. Jeffrey and Y. Onishi, The slow motion of a cylinder next to a plane wall. Q. J. Mech. Appl. Math. **34**, 129 (1981).
[34] J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez, The method of images for regularized Stokeslets, J. Comput. Phys. **227**, 4600 (2008).
[35] D. J. Smith, A nearest-neighbour discretisation of the regularized Stokeslet boundary integral equation, J. Comput. Phys. **358**, 88 (2018).
[36] D. J. Smith, A boundary element regularized Stokeslet method applied to cilia- and flagella-driven flow, Proc. R. Soc. A **465**, 3605 (2009).
[37] Lili Ju's website, https://people.math.sc.edu/ju/.
[38] H. N. Nguyen and R. Cortez, Reduction of the regularization error of the method of regularized Stokeslets for a rigid object immersed in a three-dimensional Stokes flow, Commun. Comput. Phys. **15**, 126 (2014).
[39] J. Beale, A convergent boundary integral method for three-dimensional water waves, Math. Comput. **70**, 977 (2001).
[40] P. Zheng, D. Apsley, S. Zhong, J. Sznitman, and A. Smits, Image systems for regularised Stokeslets at walls and free surfaces, Eur. J. Mech. B Fluids **97**, 112 (2023).
[41] F. Healy, H. N. Nguyen, B. Rodenborn, and O. Shindell, Trinity-Centre Google Drive Share, https://drive.google.com/drive/folders/1w-U42eVbh17HiOrBwUJB6SW0ttMy6n0s?usp=share_link